Exploring Collaborative Writing of User Stories with Multimodal Learning Analytics: A Case Study on a Software Engineering Course

RENE NOEL1, FABIAN RIQUELME1,2, ROBERTO MAC LEAN1, ERICK MERINO1, CRISTIAN CECHINEL3, THIAGO S. BARCELOS4, RODOLFO VILLARROEL5 AND ROBERTO MUNOZ1,2, (Member, IEEE)

1Escuela de Ingeniería Civil Informática, Universidad de Valparaíso, Valparaíso 2362735, Chile (e-mail: rene.noel@uv.cl, fabian.riquelme@uv.cl, erick.merino@postgrado.uv.cl, roberto.munoz@uv.cl)
2Centro de Investigación y Desarrollo en Ingeniería en Salud, Universidad de Valparaíso, Valparaíso 2362735, Chile
3Centro de Ciências, Tecnologias e Saúde, Universidade Federal de Santa Catarina, Araranguá 88906072, Brasil (e-mail: cristian.cechinel@ufsc.br)
4Laboratório de Computação Aplicada - LABCOM, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Guarulhos 07115000, Brasil (e-mail: tsbarcelos@ifsp.edu.br)
5Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362807, Chile (e-mail: rodolfo.villarroel@pucv.cl)

Corresponding author: Roberto Munoz (e-mail: roberto.munoz@uv.cl).

This work has been supported by CORFO 14ENI2-26905 “Nueva Ingeniería para el 2030” - Pontificia Universidad Católica de Valparaíso, Chile

ABSTRACT
Software engineering is the application of principles used in engineering design, development, testing, deployment, and management of software systems. One of the software engineering’s approaches, highly used in new industries, is agile development. User stories are a commonly used notation to capture user requirements in agile development. Nevertheless, for the elaboration of user stories, a high level of collaboration with the client is necessary. This professional skill is rarely measured or evaluated in educational contexts. The present work approaches collaboration in software engineering students through Multimodal Learning Analytics, modelling and evaluating students collaboration while they are writing user stories. For that, we used multidirectional microphones in order to derive Social Network Analysis metrics related to collaboration (permanence and prompting) together with human-annotated information (quality of the stories and productivity). Results show that groups with a lower productivity in writing user stories and less professional experience in managing software requirements present a non-collaborative behaviour more frequently, and that teams with a fewer number of interventions are more likely to produce a greater number of user stories. Moreover, although low experience subjects produced more user stories, a greater productivity of the most experienced subjects was not statistically verified. We believe that these types of initiatives will allow the measurement and early development of such skills in university students.

INDEX TERMS Requirements engineering, Agile development, Multimodal learning analytics, Social network analysis

I. INTRODUCTION
Requirements definition and management are considered as a critical issue in the software development process. According to a survey compiled in the Standish Group’s Chaos Report [1], software development projects that fail (i.e. the software product is not delivered) or challenge their time or budget constrains, have unsuccessful factors that include “unclear or changing requirements”, mentioned as the main cause in roughly 25% of failing and challenged projects.

The organization of the activities in the software development process is known to have an impact in the velocity and efficiency of requirement updates and also their validation by customers and stakeholders [2]. Among the possible software process models, the so-called agile methods have become very popular in the last few years. In general, agile methods
are based on the principles stated in the Agile Manifesto: (i) individuals and interactions among them should be emphasized over processes and tools; (ii) the software development process should have an emphasis on producing working software over comprehensive documentation; (iii) the customer’s active collaboration to the development team (for example, providing and validating software requirements) should be emphasized over contract negotiation in order to ensure that the software is suited to the customer’s needs; (iv) the process should incorporate practices that allow quick response to changes over ensuring that the initial plan is throughout fully followed [3].

SCRUM [4] and Extreme Programming (XP) [5] are two examples of widely adopted agile methods. The adoption of agile methods and associated practices have shown to increase the success rate of software projects and also produce software with higher quality levels, at least under specific circumstances. These results justify the teaching of agile methods to students in undergraduate courses related to Software Engineering in order to prepare these students to the practices they will most likely deal within the industry.

However, there are some challenges in training students to apply agile practices to the software development process. Some authors [6], [7] have already pointed out that the context and environment of a university course has distinguishable differences from a “real-world” software development project. Apart from the students’ lack of experience, the time dedicated to the project is usually smaller and the customer is not always available, in the cases where a real customer exists. These differences call for specific actions related to teaching and evaluation strategies.

As far as the teaching of agile methods is concerned, there are specific issues related to soft skills such as communication skills. It is known that interpersonal communication and self-organization of teams play an important role in the success of an agile project. Melo et al. [8] have identified in a multiple case-study in three software companies that inter-team communication is a core factor affecting team productivity. Hoda et al. [9] formalized the concept of self-organizing teams in agile development, meaning that the team members should play specific roles aimed at overcoming problems and self-regulating activities. The collaboration that is intrinsic to the agile process has been recognized as an important mechanism of knowledge sharing in an agile team [10]. On the other hand, it has been recognized that students often lack the collaboration and communication skills needed to achieve optimal performance when taking part in agile projects [11].

The development of communication skills has already been recognized and incorporated in didactic strategies, such as those reported by Chassidim et al. [7] and Anslow and Maurer [12]. However, assessing the development of such skills is still a challenge in real classroom environments, as the reported experiences are usually conducted in controlled environments and rely on laborious qualitative methods for analysis that do not allow rapid feedback for a teacher. In this context, Multimodal Learning Analytics (MMLA) techniques have already been employed to investigate and assess collaboration between individuals in learning environments [13]–[15], including those related to programming tasks [16].

Given the presented scenario, in this article we present the results of a case study with undergraduate students enrolled in a Software Requirements Engineering course engaged in a collaborative activity aimed at identifying and documenting software requirements as use stories [17]. The experimental hypothesis is that collaboration between individuals, as assessed by MMLA techniques, is determinant to the quality of the produced artifacts.

The goal of this study is to explore the underlying mechanisms of collaboration in a classroom environment of a highly social activity such as writing requirements in an agile context. By using MMLA techniques, we aim to characterize the collaboration among the group members, and find relationships between collaboration types, previous experience of the team members, and results of the learning activity, in terms of quality and productivity. The main results of the study indicate that, based on the number of interventions made by members of each group, in collaborative groups (i.e. where there is no a single predominant member) there is a greater proportion of individuals with higher previous experience (academic and professional) in comparison to the non-collaborative groups. In addition, inverse and strong correlations were found for experience and productivity: fewer interventions are related to better productivity and greater experience. The educational relevance of this findings yields on the insights about how the conformation and interaction of groups in a classroom setting could foster the collaboration in software engineering learning activities that which are intensive in communication skills, and in the use of MMLA for the measurement and promotion of the development of these skills.

The paper continues as follows. Section II discusses some relevant works related to Collaborative Learning and Multimodal Learning Analytics. Section III presents our used methodology and case study, namely, fifteen groups of undergraduate students interacting for the elaboration of user stories, being measured through ReSpeaker devices. In Section IV we present and discuss the main results of our experiments. Finally, Section V exposes our conclusions and future work.

II. RELATED WORK

Collaborative learning is based on the principle that knowledge is constructed through the means of social interactions, and that learning may be fostered when a group of people get together with the aim to study a given topic, to acquire new skills in some field, to solve a given problem, or to develop a common project. Collaboration may occur in groups of different scales (from pairs, to small groups or entire classes), different space dimensions (online, face-to-face, blended), different learning scenarios (informal or formal and structured), and with different periods of duration (few minutes
to several months) [18]. An example of formal structured scenarios where collaborative learning takes place are the so-called project-based learning activities, where groups of students engage to solve open-ended tasks and problems. According to Spikol et al. [19], decisive evidences about the impact of these approaches in education are still rare and may be related to the difficulties of tracking the learning processes in scenarios where the limits and boundaries are not very well determined and standardized. In this context, Multimodal Learning Analytics (MMLA) approaches can be very useful, since they allow to integrate and analyze learning traces collected from a variety of sources in order to obtain a more panoramic understanding of the learning processes [20], [21]. MMLA integrates data produced by a new generation of emerging technologies with multimodal interfaces (touch, sensors, speech, pen input, camera) allowing to follow and evaluate different dimensions related to learning, the stakeholders involved in the process, and the possible interventions [15]. MMLA allows to observe interactions and nuances that are frequently overlooked by traditional Learning Analytics techniques, since the latter normally rely only on computer-based learning contexts [22]. For instance, motion data can be used to measure the level of students attention in classroom [23] as well as to estimate students presentation skills in seminars [24]; speech information recorded during collaborative work activities can help to identify the domain expert and infer if the group answered correctly a given question [25]; and facial expressions can be used to estimate students emotional changes while they are solving exercises [26]. Collaboration is one of the constructs that has emerged across several investigations in the MMLA arena [13]. In fact, MMLA has already been applied in collaborative scenarios to investigate complex learning processes and environments, presenting interesting results that are worth to mention. One of the prominent studies on the use of MMLA to understand collaboration in the context of problem solving was done by Viatt [15]. In her experiment, students were organized in groups in order to solve geometry and algebra problems using a calculator, a digital pen and sheets of digital paper. The author collected a series of multimodal data coming from video cameras, digital pen and microphones in order to generate a number of performance metrics (time to solution, learning across sessions, cumulative expertise rating, group interaction dynamics, etc.). The experiment showed that the performance of domain experts was impressively superior from non-experts during the collaborative problem-solving sessions. They solved more problems correctly than non-experts, and contributed more to the group and in less time. A number of studies under the scope of the PELARS Project (Practice-based Experiential Learning Analytics Research And Support)1 were also developed focused on better understanding collaboration on project-based learning activities through the collection and analysis of multimodal data. For instance, Spikol et al. [19] extracted a number of multimodal features to generate models to predict successful project outcomes in open-ended tasks. Students were recorded during collaborative work focused on developing solution for specific tasks such as: prototyping an interactive toy, a colour sorted machine, and an autonomous automobile. Examples of collected features are face tracking (counting of the faces looking at the screen, distance between faces/learners), hand tracking (distance between hands, hand motion speed), audio level (sound level during the sessions) and information about the physical and software blocks used in the project (using an Arduino IDE). Data collected was pre-processed and used as input to three different parametric classifiers (Naive Bayesian, Logistic Regression, and Support Vector Machines). The results show two important features of student group work correlated to the quality of students performance in learning activities: the distance between hands, and the distance between learners. Moreover, Cukurova et al. [14] identified observable differences in students behaviors during collaborative problem solving. The authors used human observation together with students hand and head position data and were able to see that the low competent collaborative groups spent most of their time in the identification of knowledge and skill deficiencies, and little time in important aspects of problem solving (e.g. construction of hypothesis, identification of facts), whereas the most competent groups presented symmetrical contributions and high synchrony (convergence of actions) on the collaborative tasks. The authors also highlighted two important circumstances for collaborative problem solving that may be measured/observed through the implementation of their proposed coding schema: equality (whether the participants are equal in status) and mutuality (whether the participants discourse is well connected and intimate). Still under the scope of PELARS, Spikol et al. [27] indicated that student collaboration can be predicted by the distance between the learner’s hands and faces. Moreover, Grover et al. [16] captured gestures, postures, body movement and audio data from children working together (seated side-by-side) on pair-programming tasks. The collected features were used to train a Support Vector Machine classifier able to predict the level of collaboration with an accuracy of 48%. Although the findings are preliminary, the authors highlighted the potential for future assessment and scaffolding of collaboration in complex problem solving scenarios. Besides, Worsley [28] collected gesture, speech and electro-dermal activation data of 54 students while completing two engineering design tasks in pairs. The author combined the use of qualitative and quantitative techniques (human annotation, clustering, classification) to understand students learning. Among other insights, the author mentioned that bimanual coordination (active engagement) appears to correlate with learning, and that episodes of physical disengagement (planning and/or thinking) supplemented the moments of engagement. The author claims that his analysis and observations would be quite difficult to perform through non-automated means, and argues for the possible adoption of a qualitatively-informed

1www.pelars-project.eu
multimodal learning analytics approach as a mean to gain insights about educational settings. As Learning Analytics is a relatively new research field still in formation, methodological choices inside LA are continuously subject of attention and in-depth discussion [29]. In fact, a more robust methodological framework is still under construction and many researchers [30]–[34] have tried to fill this gap by structuring and studying the field according to the different dimensions LA involves (e.g. stakeholders, data sources, instruments, norms, objectives, techniques, subfields). The very nature of the field is extremely heterogeneous and methodological decisions still normally rely on the combinations of these different dimensions and specifics of each experiment. Regarding collaboration, the many works encountered in the literature also present methodological differences related to the techniques applied, the data collected, the domain of investigation, and to how collaboration is measured, analyzed and interpreted. Some of the literature addresses collaboration by generating metrics from the observation of physical interactions, while others used audio data, and/or human annotated information. Techniques revolved around supervised machine learning (classification and regression), but also used qualitative and observational approaches. As it will be described latter in Section III-B2, the present study combines the use of audio data recorded from students conversations (e.g. interactions while collaborating) combined with human annotated information (e.g. counting and rating of the resulting artifacts). Collaboration here is measured through Social Network Analysis metrics derived from students interactions and considering the speaking time of the students together with their number of interventions during the discussions. The context here is the collaborative writing of user stories during a Software Engineering Course.

III. MATERIALS AND METHODS

The following sections present the case study details according to the guidelines proposed by Runeson and Höst [35].

A. GOAL DEFINITION

The object of study are teams of students from three Software Requirements Engineering undergraduate courses. We aim to study their interactions when performing a text analysis activity to identify and document high level functional and non-functional requirements in terms of User Stories. The purpose of the study is to explore how the students interact and collaborate during the requirement analysis task, and the relationships between their previous experience and the performance and collaboration style of the group.

The research perspective is from the point of view of the researchers and teachers. The researcher or teacher would like to know if there are any systematic differences in the performance of the students in the learning activity that could be correlated to the interaction between participants. Also, the researcher or teacher would like to know if the previous background of the students in software engineering topics could be related to the way the teams organize to perform the software analysis task.

B. CASE DESIGN

1) Learning Objectives and Learning Activity

User Stories (USt) involve three types of skills which are critical for software engineers working in agile teams. In a conceptual level, USt are the basic unit for specifying “the work that may be done”, in contrast to the software requirement concept, which defines “what must be done” [5]. This conceptual difference allows the introduction of the agile vision [36] of the Iron Triangle Model for Project Managing [37], which states that the product quality is constrained by the time, cost and scope of a project: while in traditional projects the time, cost and scope are defined by contract, in agile software development the scope is variable and based on customer collaboration, being the User Stories the base of this collaborative process. From a procedural learning perspective, there is a defined structure and a set of modelling criteria that shall be met in order to correctly write a User Story [17]. Finally, attitudinal skills such as communication and team work are core principles in the agile manifesto [3], that are exercised in the process of writing and discussing User Stories [38].

The learning activity designed to achieve these learning goals was performed in two sessions of 90 minutes each (see Figure 1). The first of these two sessions was developed as follows. In the first 60 minutes the teacher covered the Agile manifesto and agile methodologies basics (principles, software development life cycle, fixed resources and fixed time planning), as well as the basic concepts and procedures for writing user stories. After that, a practical activity took place in the last 30 minutes: students were asked to conform teams of four participants, and collaboratively identify and write user stories from a real-life request for proposals (RFP) document.

The roles and tasks performed by the students were based on “User Stories Applied” by Mike Cohn [17]. However, the process was adapted considering the RFP as the source of requirements, instead of having a real stakeholder in place. Although User Stories are supposed to be written by real customers, the activity focused in the collaborative process of identifying actors or user roles, the size or complexity of the stories, and finding the business purpose for the functional or non-functional user stories. A brief feedback on these topics was presented at the end of the session.
The first session also served as a pilot for the second session in the following week, allowing the researchers to test instruments and tools which are detailed in Section III-B2. The case study address the second 90-minute session, which was aimed at applying the agile principles and requirement analysis techniques to the analysis of a new RFP document.

2) Case description and unit of analysis

Sixty undergraduate students of careers related to Computing Science, from two universities from Chile, took part in the learning activity described in the last subsection. Students, grouped in teams of four participants each, were asked to identify user stories, considering functional and non-functional aspects. The activity took 90 minutes, divided in the following steps (approximate times): 1. Activity instruction and conformation of the teams (40 minutes), 2. Review of the documents (5 minutes), 3. Solving of questions (publicly asked and answered) (5 minutes), 4. User stories writing (30 minutes), 5. Results collection and feedback (10 minutes) (see Figure 1). Figure 2 shows the execution of the activity for one of the courses.

During the step 4 of the activity, students were recorded using ReSpeaker devices, which are low-cost multidirectional microphone arrays, a four input microphone based on Raspberry Pi, capable of recording and storing voice interventions from students, also identifying which of the participants spoke at each moment, and the time duration of each intervention. The case study was made up of 15 groups \( G_i \), with \( 1 \leq i \leq 15 \), of four students each. The experiments were applied in three software engineering courses at two different Chilean universities:

- 5 groups studying Computing Civil Engineering\(^2\) (CCE) at Pontificia Universidad Católica de Valparaíso (PUCV).
- 4 groups studying Computer Engineering (CE) at PUCV.
- 6 groups studying CCE at Universidad de Valparaíso (UV).

The unit of analysis of the case is the whole group of fifteen teams conformed by sixty students that performed the learning activity. Although teams where distributed in three different Software Engineering courses, there are no significant difference in the dependent variables that could be explained by this factor, as detailed in Section III-D.

3) Case definition

According to Robson’s classification, the aim of this study is exploratory: its findings are meant to be insights to generate hypotheses for new research [39]. The setting of the case study is a real-world software engineering course, and the collection and analysis is both quantitative (measurements of performance, collaboration and clustering) and qualitative (surveys and field notes). We took a flexible approach for the design of the study: although we initially defined how to measure collaboration and the performance of the learning activity, several research questions arose only when first analysis were made, so new metrics, categories and clustering were generated to answer the research questions. We triangulated quantitative and qualitative data to ensure consistency of the measurements and observations, mainly to exclude outliers from numerical analyses based on the qualitative observations of subjects’ behavior and performance. Our main exploratory research question is ‘What is the relationship between the team collaboration and the performance in a highly social software engineering task as identifying User Stories?’ A set of detailed questions and measurements contributed to enlighten this topic:

- RQ1: What is the relationship between the productivity of the teams and the collaboration characteristics of the teams?
- RQ2: What is the relationship between the quality of the produced requirements analysis and the collaboration characteristics of the teams?
- RQ3: What is the relationship between the previous software engineering background of the team members and the collaboration characteristics of the teams?

To answer the above research questions, we considered three main data sources:

1) The ReSpeaker’s log file from the interventions made by each participant of each team. During the full duration of the activity, the four-input microphone detected when a participant talked, and a data point was added, registering a ‘1’ and the microphone which recorded the intervention. If the same participant continues talking, a stream of ‘1’s is recorded. When there is no interaction, a stream of ‘0’s is recorded. The sampling frequency of the microphone is 100 samples per second.

\(^2\)In Chile, Computing Civil Engineering is a career similar to Computing Engineering with additional management courses. It usually lasts between 5 and 6 years, in contrast to Computing Engineering, which lasts between 4 and 5 years.
per second, so we can also calculate the timestamp and duration of each intervention.

2) The cards with the user stories written by the teams, in the format ‘As a <role> I need to <requirement> in order to <goals>’. The personal survey of each participant of the team, labeled by the number of the team and the number of the microphone closer to the participant.

4) Construct Rationale and Limitations
Although the activity of requirements analysis is not methodologically representative of an agile methodology in a professional context, due to the lack of a real stakeholder, it is a collaborative learning situation in which the principles and techniques of agileism are exercised. This allows to study the collaboration of the teams solving a real problem of the discipline, such as the identification of requirements, and to explore the relationship with their performance and their previous experience in software engineering. As we shall see in Section III-C, quality is not evaluated from the point of view of the completeness or depth of the analysis, but rather of the correct writing of user stories and their relevance to the problem studied, which is an approachable learning objective within the two 90-minute sessions.

5) Selection of Subjects and Role of the Researchers
The subjects were selected for convenience, considering the availability of software engineering courses by the researchers. The participants were not assigned to teams in a random way, because this could be a distracting element to study the collaboration. As this is a case study and not an experiment (not all variables that may influence in the results are under control here), this does not threaten the validity of the study. During the learning activity, the researchers took the following roles: As a Software Engineering lecturer, one of the researchers dictated the class and ensured the correct execution of the requirements analysis activity. As experts in multimodal analysis, two researchers prepared the ReSpeakers in each classroom table and ensured that data capture was correct. As an expert in social network analysis, one of the researchers observed the interaction of the teams in order to guide the subsequent quantitative analysis of the teams’ collaboration.

6) Ethical Considerations
All participants were asked to fulfill an informed consent form, ensuring the data confidentiality. Privacy was protected by giving each participant an identification code (team number + microphone number), with which the individual surveys were labeled.

C. PROCEDURES AND DATA COLLECTION

1) Metrics and collection procedures
We defined the following metrics and measurement procedures:

1) Productivity of the teams: the number of user stories written by the team at the end of the experiment. Some teams wrote more than one story per card, so we counted the total number of stories, not the number of cards. We classified the stories as Functional and non-Functional.

2) Quality of the user stories: we assessed the quality of the user stories by rating how well the teams described the role, need and purpose of each story. The maximum rating for a story is three points (one point per section, with no decimal values). The ‘role’ section is considered well described when naming a role, a user, an actor, a position in the company, or even a software development role (only for non-functional requirements). Otherwise, the description of the section is considered as bad, i.e., if it names another subject such as a process, a system, or a hardware element. The ‘need’ section is well described if it presents a task to be fulfilled by using the system or a system property. In both cases, the task or the property must be clearly observable in the system. A bad described ‘need’ section presents a broader objective, such as ‘...need a system’ or ‘...need to reduce costs’. A well described ‘purpose’ section details a goal or a consequence of accomplishing the need. To strengthen the assessment, it was performed by two software requirement experts. Each reviewer evaluated separately the full set of stories, following the former definitions for a well defined story. We calculated Cronbach’s Alpha to measure the agreement between reviewers, finding an alpha value of 0.74, which is considered as an acceptable agreement [40].

3) Collaboration: In order to characterize the collaboration of the teams, we measured the interactions of the team members using the data log from the ReSpeaker devices. We defined an interaction as a change of participant in the intervention stream. For example, if P1 (Participant 1) is talking and then P2 talks, we count two interactions. Then, if P1, P3 or P4 talks, another interaction is registered. If a single participant talks and no other interacts (as in a monologue or a single participant reading a text), we considered only one interaction for the full length of the intervention. Moreover, note that the ReSpeaker does not detect participants speaking exactly at the same time. Hence, when it detects two sounds from different sources, it chooses the microphone with the signal with higher intensity. The ReSpeaker devices are able to measure the number of interventions of each team member, as well as the order in which the interventions were given throughout the whole activity. Besides, they can also measure the duration of each interaction, which allows to state the speaking time of each participant.

From the data collected through the ReSpeaker devices, the following measurements of collaboration where defined:
The permanence of each team member, i.e., the speaking time of a participant regarding the total duration of the conversation.

The prompting of each team member, i.e., the number of interactions or times the author received comments from another interlocutor, regarding the total number of interventions during the entire conversation.

The Collaboration type of each team was calculated from the above measures. This classification, as well as a formal definition of these measures, are shown in Section III-D1. Teams were classified as Collaborative if two or more participants were highly active, or Non-Collaborative if all team members have low interactions, or a single dominant was present.

4) Previous Software Engineering Background: each student self-reported their previous background in terms of academic and professional experience. Students were asked to classify their experience related to programming, software development and teamwork in academic context, and software development and requirement analysis in professional context, in a Likert scale of low, medium and high levels. We characterized each team of students by the median of the experience of their members, and by the level of experience of of their most expert member, for the five experience metrics.

2) Instrumentation: MMLA Software

In order to be able to visually explore the interactions between the equipment from the data of the ReSpeakers, the MMLA software was used [41]. This software system has been developed for the capture, storage, analysis and visualization of data from collaborative discussion groups [41]. Speech data is captured by multidirectional microphones (ReSpeakers), and social network analysis techniques (see Section III-D1) are used for the data analysis. Figure 3 illustrates the working scheme of the software that has been used/developed.

The developed software receives data from the microphones, which is pre-processed on Raspberry Pi devices and stored in a centralized database. Then, on the server, social network analysis techniques are used to process the data and generate the visualizations that are displayed to the client.

The main features provided are:

- Group information, or visualizations regarding a specific group. It includes visualizations for interactions between participants: precedence and intervention relationships; as well as specific visualizations for each participant: voice activation, number of interventions, voice intensity, activity and influence measures, speaking time and time duration of interventions.
- Environment information, or visualizations regarding multiple groups. It includes general visualizations for comparison between groups: group interventions, total speaking time, the total number of interactions, average voice intensity per group, the most active and influential participant of each group.

D. DATA ANALYSIS

In this section, collected data is analyzed to explore collaboration, performance and previous experience of the teams. First of all, a Social Network Analysis is presented in order to characterize collaboration of the groups based on ReSpeakers data. Then, a general view of the rest of the measurements is presented through a descriptive analysis, and finally, an exploratory analysis is performed, presenting relationships among all the measured variables to address the research questions. It is important to mention that causal relationships can not be established, since this would require an experimental approach not addressed in the present case study. The main objective of these analyzes is to identify relevant findings that can contribute to the elaboration of hypotheses about the importance of the design of teams for a collaborative learning in the account of highly social activities in software engineering.

1) Social network analysis

The interactions of each team picked up by the microphones can be represented as a social network by a multigraph \((V, E)\), where each vertex \(a \in V\) represents one of the four participants, and each edge \((a, b) \in E\) means that the sender \(a\) has transmitted a message to the team, which has been directly received by receiver \(b\), who will be the next sender to intervene in the discussion. Thus, the set of edges \(E\) represents the speech flow through time during the whole activity. Each edge has an associated timestamp that helps to modulate the entire discussion from the beginning to end.

Since the number of interventions in a collaborative work session can be very high, the multigraph can also be represented as an influence graph, without loss of information [42]. An influence graph is a labeled, weighted, directed graph \((V, E, w, f)\) representing influence relationships, where \(w : E \rightarrow \mathbb{R}\) is a weight function that assigns a weight \(w(a, b)\) to each edge \((a, b)\), and \(f : V \rightarrow \mathbb{R}\) is a label function that assigns a label \(f(a)\) to each vertex \(a\). The weight \(w(a, b)\) for edge \((a, b)\) represents the number of interventions.
issued by the sender \(a\) that were replicated by the receiver \(b\). The label \(f(a)\) for vertex \(a\) represents the speaking time of collaborator \(a\) (in seconds).

Let \((V, E, w, f)\) be an influence graph representing a collaborative group, and \(i \in V\) be a student of that group, the ReSpeaker devices allow to state the following metrics:

- \(w(a, b)\): number of interventions of student \(a\) replicated by student \(b\) (three data per student).
- \(\bar{w}(a)\): number of interventions of student \(a\) replicated by others (one data per student).
- \(f(a)\): speaking time of student \(a\) (one data per student). The pauses of the senders are not considered in the total speaking time.
- \(t(G_i)\): total time recorded for group \(G_i\) (one data per group). Although the stipulated time for the user stories identification was around 30 minutes, these times varied from one group to another, depending on the time spent in the previous steps of the activity (mainly, conformation of groups and review of documents).

The relevance of each participant in the team can be determined by using centrality measures. There are several centrality measures used to identify active, popular, or influential users [43]. However, several well known measures, such as closeness or those based on eigenvector centrality, like the PageRank [44], are not useful in collaborative networks with few nodes and many interactions. Based on the above, we define for this context two centrality measures to determine both the activity and the influence of every student within his/her group:

- The permanence of \(i\), denoted as \(Per(i)\):
  \[
  Per(i) = \frac{f(i)}{\sum_{i \in V} f(i)}
  \]

- The prompting of \(i\), denoted as \(Pro(i)\):
  \[
  Pro(i) = \frac{\bar{w}(i)}{\sum_{e \in E} w(e)}
  \]
  where
  \[
  \bar{w}(i) = \sum_{j \in V} w(i, j)
  \]

As usual, the denominators allow to normalize the measures. A student with a high permanence can be interpreted as an active individual, who may have dominated the conversation, without this implying greater influence within the group. On the other hand, a student with a high prompting can be interpreted as an idea starters [45], i.e., an influential actor who, regarding of his speaking time, usually participates repeatedly in the conversation and generates debate in the group. Note that \(Pro(i)\) is based on the classic out-degree centrality measure [46] applied to influence graphs.

In what follows, we shall denote \(Per_X(i)\) and \(Pro_X(i)\) for the permanence and prompting, respectively, of an individual \(i\) within a group \(X\).

As a first classification of the different work groups, we decided to differentiate the collaborative from the non-collaborative groups. For each group, we classify the contribution of each student in “low” (1), “medium” (2), “high” (3), using two quantiles (0.333 and 0.666) regarding to \(Per(i)\) and \(Pro(i)\). A collaborative group, with respect to \(Per(i)\) (resp. \(Pro(i)\)), is a group having two or more students with a high activity (resp. a high influence), or all the students contributing at the same level; otherwise, it is a non-collaborative group. Thus, non-collaborative groups include those with low interaction among the students, as well as those with only one dominant leader.

The classification under the permanence criterion is shown in Table 1. The third column of the table shows the speaking time \(f(i)\) in seconds of each student \(i\) within each group, as well as the total speaking time of each group. The fourth column shows the percentiles of each student \(i\), associated with their speaking time within the group. Based on these percentiles, the degree of contribution of each student is determined (fifth column), which classifies the group as collaborative (C) or non-collaborative (NC). The classification under the prompting criterion is shown in Table 2. It follows the same idea than the previous table, but instead of the speaking times, the number of interventions \(\bar{w}(i)\) are considered. The results of these tables are detailed in Section IV-A. All the different influence graphs are shown in Figure 4.

Note that, since the speaking times do not consider pauses and silences, the total speaking time for each group is always less than the total time recorded for that group, i.e., \(\text{Total}_s(G_i) < t(G_i)\) where \(t(G_i)\) is around 30 minutes for each group \(G_i\).

2) Descriptive Analysis

We characterized the experience of the teams by calculating the median of the self-reported experiences by the members (med), the maximum value of experience within the team (max), the experience in programming (prog), in teamwork (tw), in software development (swDev) and in academic contexts, the professional experience in requirements analysis (req) and in software development (swdev). The performance in the requirement analysis task was measured in the number of user stories written (Count) and the average quality assessment of the stories written by the team (Q), according to the definitions of Section III-C1. We also counted the number of Functional User Stories (F), and the number of fully
accomplished User Stories (Acc), i.e., the number of User Stories with the top quality rating (3). Table 3 summarizes the values for each group. As seen in Figure 5 the count of total user stories, functional user stories, and fully accomplished User Stories is higher in CCE-PUCV class, but the proportion between these three metrics is similar among all the three classes. Figure 6 presents the quality rating by group and by class; as can be seen, there are no significant differences among classes. Table 4 details descriptive statistics by class and by group.

Academic experience in programming, software development and teamwork are shown in Figure 7. Three classes present similar medians and error. Professional experience in software development and requirement analysis is presented in Figure 8; data suggest that self-reported professional experience is higher in CCE-PUCV class, specially for Requirement Analysis experience. Descriptive statistics for experience are detailed in Table 5.

Figure 9 presents the distribution of collaboration types of groups among classes, under the prompting criterion presented in Section III-D1. CCE-PUCV class has the higher proportion of collaborative groups (60%), while CE-PUCV class has no collaborative groups. Considering the permanence criterion, CCE-PUCV class inverts the proportion of collaborative groups (40%), while CE-PUCV and CCE-UV present a similar distribution of collaborative and non-collaborative types.

### Table 1. Speaking time (in seconds) and type of each group (*NC*: non-collaborative group; *C*: collaborative group) according to their percentile classification under permanence criterion.

<table>
<thead>
<tr>
<th>Class</th>
<th>Group</th>
<th># interventions</th>
<th>percentile</th>
<th>classification</th>
<th>Type of Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCE-PUCV</td>
<td>1</td>
<td>179.0</td>
<td>126.0</td>
<td>144.0</td>
<td>162.4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>528.0</td>
<td>201.0</td>
<td>283.0</td>
<td>1207.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>179.0</td>
<td>25.0</td>
<td>98.0</td>
<td>456.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>350.0</td>
<td>242.0</td>
<td>61.0</td>
<td>153.0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>343.0</td>
<td>275.0</td>
<td>155.0</td>
<td>80.0</td>
</tr>
<tr>
<td>CEE-UV</td>
<td>6</td>
<td>417.0</td>
<td>383.0</td>
<td>300.0</td>
<td>272.0</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>251.0</td>
<td>160.0</td>
<td>30.0</td>
<td>604.0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>372.0</td>
<td>278.0</td>
<td>144.0</td>
<td>255.0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>213.0</td>
<td>141.0</td>
<td>31.0</td>
<td>128.0</td>
</tr>
<tr>
<td>CCE-PUCV</td>
<td>10</td>
<td>428.0</td>
<td>386.0</td>
<td>494.0</td>
<td>328.0</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>507.0</td>
<td>311.0</td>
<td>475.0</td>
<td>478.0</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>682.0</td>
<td>565.0</td>
<td>367.0</td>
<td>274.0</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>603.0</td>
<td>558.0</td>
<td>490.0</td>
<td>416.0</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>538.0</td>
<td>390.0</td>
<td>384.0</td>
<td>459.0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>580.0</td>
<td>290.0</td>
<td>477.0</td>
<td>350.0</td>
</tr>
</tbody>
</table>

### Table 2. Number of interventions and type of each group (*NC*: non-collaborative group; *C*: collaborative group) according to their percentile classification under prompting criterion.

<table>
<thead>
<tr>
<th>Class</th>
<th>Group</th>
<th># interventions</th>
<th>percentile</th>
<th>classification</th>
<th>Type of Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCE-PUCV</td>
<td>1</td>
<td>179.0</td>
<td>126.0</td>
<td>144.0</td>
<td>162.4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>528.0</td>
<td>201.0</td>
<td>283.0</td>
<td>1207.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>179.0</td>
<td>25.0</td>
<td>98.0</td>
<td>456.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>350.0</td>
<td>242.0</td>
<td>61.0</td>
<td>153.0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>343.0</td>
<td>275.0</td>
<td>155.0</td>
<td>80.0</td>
</tr>
<tr>
<td>CEE-UV</td>
<td>6</td>
<td>417.0</td>
<td>383.0</td>
<td>300.0</td>
<td>272.0</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>251.0</td>
<td>160.0</td>
<td>30.0</td>
<td>604.0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>372.0</td>
<td>278.0</td>
<td>144.0</td>
<td>255.0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>213.0</td>
<td>141.0</td>
<td>31.0</td>
<td>128.0</td>
</tr>
<tr>
<td>CCE-PUCV</td>
<td>10</td>
<td>428.0</td>
<td>386.0</td>
<td>494.0</td>
<td>328.0</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>507.0</td>
<td>311.0</td>
<td>475.0</td>
<td>478.0</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>682.0</td>
<td>565.0</td>
<td>367.0</td>
<td>274.0</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>603.0</td>
<td>558.0</td>
<td>490.0</td>
<td>416.0</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>538.0</td>
<td>390.0</td>
<td>384.0</td>
<td>459.0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>580.0</td>
<td>290.0</td>
<td>477.0</td>
<td>350.0</td>
</tr>
</tbody>
</table>
collaborative groups, as shown in Figure 10.  

3) Exploratory Analysis

We explored the collected data to address the three research questions presented in Section III-B3. Main findings are commented below.  

RQ1: What is the relationship between the productivity of the teams and the collaboration characteristics of the teams? We compared medians and distribution for the productivity variables Count (number of user stories written), F (number of Functional User Stories), grouping by collaboration types. No relevant differences among groups were observed for the permanence criteria. For the biggest differences between groups were for population distribution of (Count) and (F). As shown in Figures 11 and 12, more non-collaborative groups produced less than five User Stories and Functional User Stories than the collaborative groups.

Also, we correlated the number of interventions of each team with the productivity. As seen in Figure 13, a negative correlation with the Number of User Stories and Number of Functional User Stories can be seen. We performed a correlation analysis for Number of User Stories. We found that both variables were normally distributed (assessed by Shapiro-Wilk’s test with p > 0.05), Pearsons’s correlation showed a statistically significant, strong negative correlation \( r(13)=-0.723, p=0.002 \). Teams with a fewer number of intervention are more likely to produce a greater number of user stories. No significant correlation was found for Number of Functional User Stories.

RQ2: What is the relationship between the quality of the
requirements analysis of the teams and the collaboration characteristics of the teams? Considering the quality rating \((Q)\) and the fully-accomplished User Stories \((Acc)\) variables, no differences are shown between collaborative and non-collaborative groups under permanence and prompting criteria, neither for population distribution nor for the median measures. We also tested the association for the number of interventions and the quality rating of the user Stories, but as seen in Figure 14, no linear correlation can be stated.

**RQ3:** What is the relationship between the previous software engineering background of the team members, and the collaboration characteristics of the teams?

For academic experience, no relevant differences were found under the permanence and prompting criteria of collaboration. On the other hand, self-reported professional experience seems to be clearly different between collaborative and non-collaborative groups under prompting criteria. As shown in Figures 15 and 16, medians and maximum professional experience in requirements analysis are not equally distributed among collaboration types. There is a greater number of groups of students with little experience in the group of non-collaborative groups, with respect to the set of collaborative groups, both for the median experience of the student group and for the maximum level of experience in the group.

Looking for the association of academic experience and collaboration, we applied Spearman’s non-parametric correlation test due to the not normal distribution of experience variables. Academic experience in software development is significantly correlated with the number of interventions, both for the median \((r=-0.652, p=0.008)\) and maximum \((r=-0.788, p=0.000)\) measurements. As shown in Figure 17, a higher experience in software development in an academic context is associated with fewer interventions.

Finally, for the association between professional experience and the number of interventions, Requirements Analysis experience was present an statistically significant strong, negative correlation with the number of intervention, both for median \((r=-0.622, p=0.013)\) and maximum \((r=-0.527, p=0.044)\) measurements, according to Spearman’s correlation. Depicted in Figure 18, the association suggests that a higher professional experience in Requirements Analysis is associated with a fewer number of interventions.

**IV. RESULTS AND DISCUSSION**
A. SOCIAL NETWORK RESULTS

From Table 1 we conclude that the permanence criterion classified the groups in an equitable way: 7 of 15 groups (47%) as collaborative, and the remaining 8 (53%) as non-collaborative. Instead, the prompting criterion illustrated in Table 2 classified just 5 of 15 groups (33%) as collaborative, and the remaining 10 (67%) as non-collaborative. Indeed, in groups 8 and 9, although there were two students who dominated the conversation with a high speaking time ($Per_{8}(1) = 0.932$, $Per_{8}(2) = 0.695$, $Per_{9}(1) = 0.780$, and $Per_{9}(2) = 0.814$), only one of the two students from each group managed to exert a greater influence on the group, receiving greater feedback ($Pro_{8}(1) = 0.881$ and $Pro_{9}(1) = 0.949$). However, this does not mean that the prompting criterion always classifies more non-collaborative groups than the permanence criterion. In effect, in group 4, student 1 totally dominated the conversation ($Per_{4}(1) = 1.000$) but student 2 also had a lot of feedback, becoming an influential actor as well ($Pro_{4}(2) = 0.763$).

B. COLLABORATION, PERFORMANCE AND EXPERIENCE RESULTS

Comparing the prompting and permanence criteria for types of collaboration of the groups, the relationships found for research questions 1 and 3 suggest that prompting criteria yields to a better differentiation of the collaboration behavior of the teams. In this case, productivity and prior experience seems to be clearly distinguished by prompting. Under this criteria, it seems that groups with a lower productivity of user stories and professional experience in software requirements, present a non-collaborative behaviour more frequently. These results suggest that less experienced subjects tend to conform non-collaborative teams, and non-collaborative teams are less productive than collaborative teams, considering the professional experience of its members in requirements analysis.

The above affirmation can be supported by comparing the productivity of groups with different levels of experience in requirements analysis: as show in Figure 19, groups in which its member with maximum level of expertise has no experience in software requirement (value=1), produce fewer user stories, with statistical significance of $p= 0.23$, for Independent Samples Median Test.

This result is relevant for supporting the Collaboration by Prompting as a useful characterization for analyzing the behaviour of teams performing highly social software engineering learning tasks. Further research (probably with an experimental approach) is required to assess causality...
between collaboration type by prompting and productivity, mitigating the bias in the conformation of the groups due to prior experience.

The prevalence of the prompting criteria also lead us to study the relationships for performance and experience variables with the total number of interventions. Inverse, strong correlations were found for experience and productivity: smaller number of interventions are related to a better productivity and a higher experience. Although no causal relationships can be stated, a potential hypothesis that can be drawn is that more experienced subjects tend to interact less, but produce more user stories. This is an intuitive result, which can be associated to professional skills such as a higher concentration power, favored by the custom of performing the task entrusted, or to the spontaneous emergence of a leader (the more experienced group member). Also, a high number of interventions could be interpreted as more trivial conversation interactions, instead on working in the learning activity; however, field observations do not confirm this suspicion.

Above results seem to be not related with the quality of the results, considering “a well written” User Story as the quality criteria. Neither collaboration types (under both prompting and permanence criteria), nor experience seems to be related to the quality of the user stories.

One hypothesis to explain the lower productivity of teams with less previous experience and a high number of interventions is that students might have spent more time trying to know what to do, than collaborating to write user stories. For the greater predominance of teams with low experience within the non-collaborative teams, we believe that the low average experience of the entire team or the lack of knowledge about how to carry out the task could give rise to the emergence of natural leaders who have monopolized the interventions, which contrasts to the idea of horizontal and self-organized agile teams [9].

These results should be considered within the constraints of the present study. The duration of the activity that we could measure with ReSpeaker (30 minutes) may not be enough to notice important differences in quality between user stories. On the other hand, although ReSpeaker allows recording of the audio content, as far as the analysis of the content of the recorded discussions is concerned, the quality of the obtained audio did not allow to extract this information. It should also be considered that the ambient noise of the classroom can introduce wrong measurements, but based on our observation of the activity, we do not believe that
there are large differences. At the methodological level, the experience was self-reported through an ad-hoc instrument, which could be improved with an entrance test to characterize the participants, in future investigations with an experimental approach.

V. CONCLUSIONS

As seen in the previous section, main findings of the case study are the relationships between prior experience in software requirements and the way the team members collaborate, and the lower productivity of low experienced groups. This result seems to be consistent with the findings of [14], where low competent collaborative groups spend most of their time in the identification of knowledge and skill deficiencies, and little time in important aspects of problem solving, whereas the most competent groups presented symmetrical contributions and high synchrony (convergence of actions) on the collaborative task. However, no evidence was found that can relate to [15], where performance of domain experts was impressively superior from non-experts during collaborative problem-solving sessions. Although it was stated that low experience subjects produced more user stories, a greater productivity of top experience subjects was not statistically verified.

Considering the rules for classifying collaboration types, prompting criteria allowed a better differentiation among groups, and relevant correlations were found for the number of interventions, in spite of total speaking time and permanence criteria. This finding may allow to improve the results presented in [47], were the rules that have considerable power in identifying extreme cases of collaboration were: 1) non-collaborative blocks were characterized by low level of talk, asymmetry in the conversation and high levels of physical action; 2) collaborative blocks were characterized by high levels of symmetric conversation and less physical interactions with the system.

Future work is associated with two areas: from the perspective of MMLA, the improvement of the ReSpeaker’s capabilities is considered in order to increase the recording time of the activity, ideally delivering a continuous flow of the collected data for analysis and real-time deployment of collaboration indicators. On the other hand, the present investigation will continue with the design and execution of experiments that allow verification of causal relations between the collaboration, measured in terms of number of interventions and classified according to the prompting criteria, and the quality and productivity in writing user stories, considering the previous experience of the participants as an influence factor. Improvements in the process of quality assessment of user stories will also be considered, considering not only the writing format, but also relevant features for requirement analysis such as consistency, uniformity and independence.

REFERENCES


RENE NOEL is an associate professor at the School of Computer Engineering of the Universidad de Valparaíso. His research interests are agile software development, empirical software engineering and engineering education. René has led several software development projects in social areas such as education, assistive technologies and health information systems. Also, René has a Master’s degree in computer science and education, and is a research associate at the National Center for Health Information Systems, CENS.
FABIÁN RIQUELME is an associate professor at School of Informatics Engineering and responsible researcher at the Centro de Investigación y Desarrollo en Ingeniería en Salud (CINGS) of Universidad de Valparaíso, Chile. Fabián received the bachelor in Computing Civil Engineering and a master in Computer Science from Universidad de Concepción, Chile, and a PhD in Computing from Universitat Politécnica de Catalunya, Barcelona, Spain. In his PhD thesis, he proposed the influence graphs as a model to study influence spread in social systems. His main research interests are social network analysis, social computing, cooperative game theory and computational complexity.

ROBERTO MAC LEAN is a Computer Engineer (2018) at Universidad de Valparaíso. Roberto’s Thesis was about analyzing and visualizing the students’ behaviors during non-structured activities in educational environments. Currently, Roberto works at The National Center for Health Information Systems (abbreviated CENS in Spanish). His research interests are focused on Multimodal Learning Analytics and Health Informatics.

ERICK MERINO is a Computer Engineer (2018) at Universidad de Valparaíso. Erick’s Thesis was about analyzing and visualizing the students’ emotions via EEG in educational environments. Currently, Erick has enrolled in the Master in Biomedical Engineering program at the Universidad de Valparaíso. His research interests are focused on Multimodal Learning Analytics and Big Data.

CRISTIAN CECHINEL is an assistant professor at the Sciences, Technologies and Health Center (CTS), Federal University of Santa Catarina (Araranguá), Brazil. Cristian received the bachelor’s and master’s degrees in Computer Science from the Federal University of Santa Catarina in 1998 and 2000, respectively, and the Ph.D. degree in Information and Knowledge Engineering from the Computer Science Department, University of Alcalá, Spain, in 2012. His research mostly focuses on the development and analysis of digital learning technologies, learning analytics and distance learning. He is an Active Member of the Latin-American Community on Learning Technologies (LACLO) and a former member of the Special Committee on Computers and Education of the Brazilian Computer Society (CEIE-SBC). He is also an Associate Editor of the Brazilian Journal of Computers in Education (2016-2018).

THIAGO S. BARCELOS is an associate professor at the São Paulo Federal Institute of Education, Science and Technology. Thiago received his bachelor and master degrees in Computer Science from the University of São Paulo in 2002 and 2005, respectively. His PhD degree in Science and Mathematics Teaching was obtained in 2013 from Cruzeiro do Sul University, São Paulo. In his PhD thesis, Thiago investigated collaborative approaches based on building digital games for the development of Computational Thinking skills. He is currently a member of the Special Committee on Computers and Education of the Brazilian Computer Society (CEIE-SBC) and holds a position as Associate Editor of the International Journal on Computational Thinking (IJCThink). His research interests are focused on Computers and Education, Agile Methods and Multimodal Learning Analytics.

RODOLFO VILLARROEL is an associate professor and researcher at the School of Computer Engineering of the Pontificia Universidad Católica de Valparaíso, Chile. Professor Villarroel obtained his Ph.D. degree at the University of Castilla-La Mancha (2005), Ciudad Real, Spain, and his Master’s Degree in Computer Science Engineering at the Universidad Técnica Federico Santa María (2000), Valparaíso, Chile. He is currently a board member of the Chilean Computer Science Society (SCCC). His research areas are: Software Engineering, Quality and Information Security, and Computers and Education.

ROBERTO MUNOZ is an associate professor of the School of Informatics Engineering and adjunct researcher at the Center of Cognition and Language (CIDCL) and at the Centro de Investigación y Desarrollo en Ingeniería en Salud of Universidad de Valparaíso. Professor Munoz holds doctoral studies in Computer Engineering, as well as Masters in Computer Engineering, Engineering Science, and Education. He is the author of over 50 scientific papers in refereed international conferences and journals. Also, professor Munoz holds a position as Editor of the International Journal on Computational Thinking (IJCThink). His research areas are focused on Computers and Education, Multimodal Learning Analytics, Human-Computer Interaction, and Health Informatics.

***